Estimation of nonparametric dynamical models within Reproducing Kernel Hilbert Spaces for network inference

Florence d'Alché-Buc[1,2]

Joint work with Nehemy Lim (IBISC/CEA), George Michailidis (Michigan U.) and Yasin Senbabaoglu (Michigan U.)

[1]Visiting: INRIA, LRI CNRS 86 23, Université Paris-Sud, Orsay, France
[2]Permanent address: IBISC EA 4526, Université d'Évry Val d'Essonne, Évry cedex, France

June 6, 2012, PEDS II

Reverse-engineering of biological networks

- Identify and understand complex regulatory mechanisms at work in the cell
- Gene regulatory networks: nonlinear dynamical systems whose states are concentrations of mRNA's
- Two main tasks:
  - Modeling and parameter estimation
  - Network inference
- Our Goal: from experimental data and prior knowledge use statistical inference to unravel biological networks

Network inference

- **Data**: time course of gene expression, steady states, perturbation data (gene regulatory networks)
- Two big families of approaches:
  - Edge prediction:
    - Supervised approaches: (Qian et al. 2003, Yamanishi et al. 2004, Ben-Hur and Noble 2005, Geurts et al. 2006,07, Yamanashi and Vert 2007, Bleakley et al. 2007, Kashima et al. 2009, Yip and Gertsein 2009,)
    - Semi-supervised approaches: Tusda et al. 2003, Kashima et al. 2009, Brouard et al. 2011
  - Modeling the network:
    - Static models: bayesian networks (Friedman et al. 2001, Husmeier and Vehrli 2007, Auliac et al. 2008), sparse models (Schaffer and Strimmer et al. 2005, Meinhausen and Bulhman 2006) ...
    - Dynamic models: next slide

Network inference from time series

- **Measurements of coupling (Kramer at al. 2009)**
- **Sparse linear models**
  - Autoregressive models (Opgen-Rhein and Strimmer, 2007)
  - Granger causality (Shojaie and Michailidis, 2010)
  - State-space models (Perrin et al. 2003, Rangel et al. 2004)
- **Nonlinear parametric models**
  - S-systems (Chou et al. 2006, Vilela et al. 2008)
- **Nonlinear nonparametric models**
  - Inferrelator (Bonneau et al. 2006)
  - Dynamic Bayesian Networks (Imoto et al. 2002, Husmeier et al., Li et al. 2007, Bansa et al. 2007)
  - Gaussian processes for network inference (Aijö and Lähdesmäki 2009)

Motivation for our approach

- Extend linear approaches to network inference to nonlinear models
- Generic approach to network inference based on nonparametric modeling
- Should work with and without prior knowledge about the graph
- Should scale to large networks

Modeling dynamical system

- We consider two frameworks
  - Network inference with nonparametric autoregressive models
  - Network inference with ODE and generalized profile estimation method

Autoregressive models

- $x_{t+1} = h_{true}(x_t) + \epsilon_{t+1}$
- ϵ_{t+1}: iid (Gaussian) noise term
- Build a nonparametric estimate h_{N-1} from $x_{t_0}, \ldots, x_{t_{N-1}}$
- Solve the problem in the context of regularization theory

Network inference with nonparametric modeling

Use the Jacobian

- Build a nonparametric estimate h_{N-1} from the observed time-serie $x_0, \ldots, x_{N-1}$
- Compute the empirical mean of the Jacobian:

$$\hat{A}_{ij} = \theta\left(\sum_{t=0}^{N-1} \frac{\partial h_N(x_t)_i}{\partial (x_t)_j}\right), \tag{1}$$

$\theta : \mathbb{R}^p \to \{-1, 0, 1\}$ is a thresholding function.

Nonparametric network inference

- This jacobian-based network inference method applies to any model
- But we propose here a new kernel-based model that is able to incorporate prior information and that has an interesting jacobian
- We explain the approach in the case of autoregression and show how it can be applies in generalized profile estimation

Nonparametric autoregressive model

Operator-valued kernel vector autoregressive model

- Given $\mathcal{S}_N = \{x_0, \ldots, x_{N-1}\}$ (we note x_i for x_{t_i})

$$h(x_t; \mathcal{S}_N) = \sum_{\ell=0}^{N-2} K(x_\ell, x_t).c_\ell \tag{2}$$

- K: $\mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^p$ is an operator-valued kernel
- $c_\ell \in \mathbb{R}^p, \ell = 0, \ldots, N-2$

Outline

Outline

Notations and definition

Let $\mathcal{X}$ be a set and $\mathcal{F}_y$ a Hilbert space. Denote by $L(\mathcal{F}_y)$, the set of all bounded linear operators from $\mathcal{F}_y$ to itself. Given $A \in L(\mathcal{F}_y)$, A^* denotes its adjoint.

Definition (Operator-valued kernel [8, 2])

Let $\mathcal{X}$ be a set and $\mathcal{F}_y$ a Hilbert space. Then, $K : \mathcal{X} \times \mathcal{X} \to L(\mathcal{F}_y)$ is a kernel if:

- $\forall \ (x, y) \in \mathcal{X} \times \mathcal{X}, K(x, y) = K(y, x)^*$
- $\forall m \in \mathbb{N}, \forall \{(x_i, y_i)\}_{i=1}^m \subseteq \mathcal{X} \times \mathcal{F}_y, \sum_{i,j=1}^m \langle y_i, K(x_i, x_j) y_j \rangle_{\mathcal{F}_y} \geq 0$

Example

Kernel for multi-tasks learning [6] and structured classification [3]

$\mathcal{X} = \mathbb{R}^p$ and $\mathcal{F}_y = \mathbb{R}^d$ (d classes or d tasks). Decomposable kernel based on the product of a scalar kernel and a semi-definite positive matrix B of size $d \times d$ has been proposed:

$$\forall (x, z) \in \mathbb{R}^p \times \mathbb{R}^p, K(x, z) = k(x, z)B \tag{3}$$

where B encodes the dependence between tasks or between classes.

Building a RKHS from K

Theorem

(Senkene and Tempel'man 1973, Micchelli and Pontil 2005) Let $\mathcal{X}$ be a set and $\mathcal{F}_y$ be a Hilbert space. If $K : \mathcal{X} \times \mathcal{X} \to L(\mathcal{F}_y)$ is an operator-valued kernel, then there exists a unique RKHS $\mathcal{H}_K$ which admits K as the reproducing kernel; that is

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{F}_y, \forall h \in \mathcal{H}\ \langle h, K(x,\cdot)y\rangle_{\mathcal{H}} = \langle h(x), y\rangle_{\mathcal{F}_y}\,. \quad (4)$$

The RKHS $\mathcal{H}_{\mathcal{K}_x}$ is built by taking the closure of $span\{K(x,\cdot)\mathbf{y} | x \in \mathcal{X}, \mathbf{y} \in \mathcal{F}_y\}$ endowed with the scalar product $\langle f, g\rangle_{\mathcal{H}_K} = \sum_{i,j}\langle u_i, K(r_i, s_j)v_j\rangle_{\mathcal{F}_y}$ with $f(\cdot) = \sum_i K(r_i,\cdot)u_i$ and $g(\cdot) = \sum_j K(s_j,\cdot)v_j$. The corresponding norm $\|\cdot\|_{\mathcal{H}_K}$ is defined by $\| f \|^2_{\mathcal{H}_K} = \langle f, f\rangle_{\mathcal{H}_K}$.

Representer theorem

Theorem ([6])

Let V be a convex loss function, and $\lambda > 0$ the regularization parameter. Let $\{(x_i, y_i), i = 1, \ldots, N$ be the set of data. Then, the minimizer of the following optimization problem:

$$argmin_{h \in \mathcal{H}} \mathcal{L}(h) = \sum_{i=1}^{N} V(h(x_i), \mathbf{y}_i) + \lambda \|h\|_{\mathcal{H}}^2 \ ,$$

admits an expansion:

$$\hat{h}(\cdot) = \sum_{\ell=1}^{N} K(x_\ell, \cdot) c_\ell \ , \tag{5}$$

where the coefficients $c_\ell, \ell = \{1, \cdots, N\}$ are vectors in the Hilbert space $\mathcal{F}_y$.

Outline

Operator-valued kernel-based vector autoregressive model

OKVAR

- $\mathcal{X} = \mathcal{F}_y = \mathbb{R}^p$.
- Solve the task as a regression one with $\mathcal{S}_N = \{(x_0, x_1), (x_1, x_2), \ldots, (x_{N-2}, x_{N-1})\}$,
- If we choose a loss function $\mathcal{L}$ such as in the previous theorem, the following model is justified by the representer theorem

$$h(\mathbf{x}_t; \mathcal{S}_N) = \sum_{\ell=0}^{N-2} K(x_\ell, x_t).c_\ell \qquad (6)$$

- K: $\mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^p$ is an operator-valued kernel
- $c_\ell \in \mathbb{R}^p, \ell = 0, \ldots, N-1$
- More generally, we choose this model based on a finite expansion on the functions $K(x_\ell, \cdot)$ even if it is not justified by a representer theorem.

Which matrix-valued kernel? (1)

$\mathcal{X} = \mathcal{F}_y = \mathbb{R}^p$ Transformable kernels (Caponnetto et al.). Let k be a scalar (sdp) kernel defined on $\mathbb{R} \times \mathbb{R}$:

$$\forall \ (x, y) \in \mathbb{R}^p \times \mathbb{R}^p, K_k(x, y)_{ij} = k(x^i, y^j) \tag{7}$$

K_k is a matrix-valued kernel. x^i i-th coordinate of x. Proposition. Let B a semi-definite positive matrix of size $p \times p$ and $T_q : \mathbb{R}^p \to \mathbb{R}$, the projection on the q-th dimension. The kernel defined by

$$\forall \ (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^p \times \mathbb{R}^p, K_{kB}(x, y) = B \circ K_k(x, y) \tag{8}$$

is a matrix-valued kernel, where $\circ$ denotes the Hadamard product for matrices.

Which matrix-valued kernel? (2)

- Let us take k as the Gaussian kernel:
- $\forall \ (x, y) \in \mathbb{R}\mathbb{R}, k(x, y) = exp(-\gamma(x-y)^2)$.
- The model writes as:
- $h(x_t; \mathcal{S}_N) = \sum_{\ell=0}^{N-2} B \circ K_{Gauss}(x_\ell, x_t).c_\ell$

Jacobian of the B-decomposable kernel

$$J_{ij}(t) = 2\gamma b_{ij} \sum_\ell (x_\ell^i - x^j(t)) exp(-\gamma(x_\ell^i - x^j(t))^2) c_{\ell j}.$$

Outline

Learning h when B is fixed

1. B is given prior to the data
2. B is estimated from the data independently from C

Role of B

Properties

- B is sdp
- $\forall (i,j) \in \{0,\ldots,p\}^2$ if $b_{ij} = 0$, then $J_{ij} = 0$

B acts as a mask on the kernel K

- We propose to define B as the Laplacian of some matrix W
- B = D - W
- W is symmetric binary matrix, defining a mask on the network

How to choose W ?

1. If we know the network structure in terms of a non-oriented adjacency matrix, $W = W_{true}$: the issue of network inference reduces to find the signed edges
2. If we know that the network is a modular one, for instance composed of m modules, we can provide a block diagonal matrix W, with 1-valued blocks
3. If we only know that some of the regulatory edges are not possible, e.g. j cannot be a regulator of i and i cannot be a regulator of j, we can impose a zero for b_{ij} and 1 everywhere else
4. If we have not prior knowledge about the structure, we can fix $W = 1_H$ where 1_H is the neutral element for Hadamard product.

How to learn W, before learning C ?

- Linear partial correlation: Approximate W by using empirical precision matrix: $W \equiv \theta((\Sigma_{N-1})^{-1})$
- Nonlinear independence test: using Hilbert-Schmidt independence criterion

Learning C, given W

Elastic loss $\mathcal{L}_{elastic}$ (elastic net: scalar case Zhou and Hastie)

- the ℓ_1 constraint useful for: (1) high dimensionality and (2) sparsity of the jacobian

$$\mathcal{L}_{1,2}(C) = \sum_{t=0}^{N-2} \|x_{t+1} - h_C(x_t)\|^2 + \lambda_2 \|h_C\|_{\mathcal{H}}^2 + \lambda_1 \|C\|_1 \qquad (9)$$

- Implementation:
  - $\lambda_1 = 0$: closed formed solution as in the scalar case: $C = (\mathcal{K} + \lambda_2 Id)^{-1} Y$ with Y vector column of $x_0, \ldots, x_{N-2}$, and C is the corresponding vector column.
  - $\lambda_1 > 0$: subgradient methods

When $\frac{N}{p} < 1$

- We have to learn C, a matrix of size $N \times p$
- To cope with large p (large networks), we reduce the dimension and work on submatrices (subnetworks)
- An ensemble learning strategy : boosting [Freund and Schapire 96, Friedman et al. 2000, Mason et al. 2000]
- Boosting is a stage wise algorithm to build an additive model
- L_2 Boosting [Friedman et al. 2000] : add a new base model to fit the residuals

Boosting algorithm

1. $\forall t \in \{0,\dots,N-1\}$, $H_0(\mathbf{x}_t) := (\bar{\mathbf{x}}^1,\dots,\bar{\mathbf{x}}^p)^T$ and $\mathbf{u}_t^{(0)} := \mathbf{x}_t$
2. Iteration $m=0$, STOP=false
3. WHILE $m < M$ and STOP=false
   (a) Update $m \leftarrow m+1$
   (b) Select $\mathcal{S}_m$, a random subset of genes of size $k \leq p$
   (c) Compute the residuals $\mathbf{u}_{t+1}^{(m)} := \mathbf{x}_{t+1} - H_{m-1}(\mathbf{x}_t)$
   (d) STOP := true if $\forall j \in \{1,\dots,p\}$, $\|\mathbf{u}^{j\,(m)}\| \leq \epsilon$
   (e) IF STOP=false
      - (a) Estimate $W_m \in \{0,1\}^{k\times k}$ from $\mathbf{u}_1^{(m)},\dots,\mathbf{u}_N^{(m)}$
      - (b) Estimate the parameters C_m with the elastic model
      - (c) Update the m^{th} boosting model: $H_m(\mathbf{x}_t) := H_{m-1}(\mathbf{x}_t) + \rho_m h(\mathbf{x}_t; \{\mathcal{S}_m, W_m, C_m\})$
4. ENDWHILE
5. $m_{stop} := m$
6. Compute the Jacobian matrix $J_{m_{stop}}$ of $H_{m_{stop}}$, threshold and get $\hat{A}$.

Outline

Collocation, generalized profile estimation

Principle

- Refs: [Varah, 82], [Tjoa and Biegler 91], [Ramsay et al. 2007], [Brunel 2008], [Brunel and d'Alché-Buc 2010],[Gugushvili and Klaassen, 2010],[Cao et al, 2011]
- Approximate the observed trajectory x(t) by a basis expansion function and use the derivative of this function in the square loss to estimate parameters of the ODE
- Iterative scheme to improve upon the two-steps scheme

ODE, generalized profile estimation method and nonparametric approach

ODE

- $\dot{x} = h_{true}(x(t))$
- $x(0) = x_0$
- Noisy observations: $y(t_\ell) = x(t_\ell) + \epsilon(t_\ell), \ell = 0, \ldots, N-1$
- We propose to build two nonparametric estimates:
- g_{N-1}, estimate of x and h_{N-1}, estimate of h_{true} from $x_0, \ldots, x_{N-1}$

RKHS everywhere !

RKHS $\mathcal{G}_G$

Given $\mathcal{S}_N = \{y_0, \ldots, y_{N-1}\}$, G: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}^p$, an operator-valued kernel:
We search for $g \in \mathcal{G}_G$ satisfying:
$g(t) = \sum_{\ell=0}^{N-1} G_\sigma(t_\ell, t)\beta_\ell$
and being an estimation of the solution of the ODE

RKHS $\mathcal{H}_K$

Given $\mathcal{S}_N = \{y_0, \ldots, y_{N-1}\}$, K an operator-valued kernel, we search for $h \in \mathcal{H}_K$, a nonparametric model of the true ODE model f, with the following form:

$$h(\cdot) = \sum_{\ell=0}^{N-2} K_{\gamma,W}(y_\ell, \cdot)\alpha_\ell \qquad (10)$$

Smoothing-based profiled estimation for network inference: loss function

We want to minimize:

$$\mathcal{L}(g, h; x_0^{N-1}) = \sum_{\ell=0}^{N-1} V_g(g(t_\ell), y_\ell)) + \lambda_g \Omega_g(g) + \dots$$

$$\lambda \int V_{g,h}(\dot{g}(t), h(g(t)))dt + \lambda_h \Omega_h(h)$$

Once we get (g_N, h_N), we can extract $\hat{A}$ from the mean Jacobian

Smoothing-based profiled estimation for network inference

Algorithm

1. m = 0
2. Learn g_m from $\mathbf{x}_0, \cdots, \mathbf{x}_{N-1}$
3. STOP = false
4. While stopping criterion = false DO and $m < MAX$
   - $m = m + 1$
   - Minimize (g_{m-1}, h_m) according to h_m, g_{m-1} being fixed
   - Minimize (g_m, h_m) according to g_m, h_m being fixed
   - Compute STOP, the stopping criterion
5. $m_{stop} := m$
6. Compute the Jacobian matrix $J_{m_{stop}}$ of $h_{m_{stop}}$, threshold and get $\hat{A}$.

Outline

Experimental results

- Artificial data: DREAM3
  - data generated by imbuing the networks with dynamics from a thermodynamic model of gene expression and a Gaussian noise ([7])
  - networks are subgraphs of the currently accepted *E. coli* and *S. cerevisiae* gene regulation networks, exhibiting various patterns of sparsity and topological structure.networks of size 10 and 100 [7]

DREAM3: properties of the graphs

Size10	Ecoli1	Ecoli2	Yeast1	Yeast2	Yeast3
Average degree	2.2	3.0	2.0	5.0	4.4
Density	0.244	0.333	0.222	0.556	0.489
Modularity	0.016 (2)	0 (1)	0.260 (3)	0 (1)	0 (1)
Size100	**Ecoli1**	**Ecoli2**	**Yeast1**	**Yeast2**	**Yeast3**
Average degree	2.5	2.38	3.32	7.78	11.02
Density	0.025	0.024	0.033	0.079	0.111
Modularity	0.643 (6)	0.661 (7)	0.681 (8)	0.328 (6)	0.088 (14)

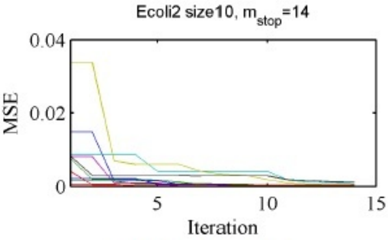
DREAM3-size10 (1)

	Ecoli1 $\lambda_1 = 1,\ \lambda_2 = 10^{-4}$		Ecoli2 $\lambda_1 = 1,\ \lambda_2 = 10$	
W	AUROC	AUPR	AUROC	AUPR
Truth	0.8989	0.3493	0.8529	0.5566
Partial Corr.	**0.7125**	**0.2492**	0.6176	0.3702
1_H	0.6609	0.1677	0.6431	**0.4152**
Team 236	0.621	0.197	**0.650**	0.378
Team 190	0.573	0.152	0.515	0.181

DREAM3 (2)

	Yeast1 $\lambda_1 = 1,\ \lambda_2 = 10^{-1}$		Yeast2 $\lambda_1 = 10^{-1},\ \lambda_2 = 10^{-2}$		Yeast3 $\lambda_1 = 1,\ \lambda_2 = 10^{-1}$	
W	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Truth	0.8944	0.4754	0.7733	0.3368	0.8077	0.3646
PC	0.6444	**0.3220**	0.6067	0.2355	**0.6416**	0.3415
1_H	0.5889	0.2733	**0.6600**	**0.3186**	0.6206	**0.3421**
Team 236	**0.646**	0.194	0.438	0.236	0.488	0.239
Team 190	0.631	0.167	0.577	0.371	0.603	0.373

DREAM3 (boosting [5])



(a) Size10 Ecoli2

DREAM3 (boosting)

Results with another kernel: $K(x, y)_{ij} = exp(-\gamma_{ij}(x^i - x^j)^2)$
Base learner is with W taken as the empirical precision matrix

	Ecoli1		Ecoli2		Yeast1		Yeast2*		Yeast3*	
Size 10	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Base learner	0.614	0.226	0.616	0.406	0.578	0.343	0.580	0.233	0.612	0.300
Multiple-run DynBoost	**0.654**	**0.301**	0.718	0.659	0.628	0.160	0.720	0.352	0.684	**0.525**
Bootstrap DynBoost	0.620	0.241	**0.923**	**0.876**	**0.722**	**0.393**	**0.733**	**0.386**	**0.695**	0.359
Team 236	0.621	0.197	0.650	0.378	0.646	0.194	0.438	0.236	0.488	0.239
Team 190	0.573	0.152	0.515	0.181	0.631	0.167	0.577	0.371	0.603	0.373

Conclusion and on-going work

- Nonparametric estimation for network inference: promising results
- Regularization framework / MAP approach (although MAP not used *per se* here)
- *Different theoretical issues*:
  - Consistency results (on-going work)
  - Learning/optimization of B and C (B sdp)
  - More sophisticated constraints on C

Conclusion and on-going work

Practical issues

- Model selection (GV extended to ridge and hinge losses)
- Large networks: modular approaches
- Other kinds of data : steady-state, perturbation data
- Application to RNQ-Seq data: response to retinoic acid (with Gerard Benoit, INSERM,Lyon)
- July: ODESSA-lib, a matlab library

Two postdoc positions / One workshop

- Protein-protein interaction network prediction (CFTR network, cystic fibrosis) Two postdoc positions / One workshop
- Dynamical modeling for understanding of endothelium dysfunctions in normal tissues following ionizing radiation exposure with Olivier Guipaud (IRSN, Paris)
- **One workshop at ICML 2012**:
  - Object, functional and structured data: towards next-generation kernel methods
  - June 30, 2012. Edinburgh, UK.
  - `icml.cc/2012/workshops`
  - Contact: florence.dalche AT ibisc.fr or florence.dalche AT inria.fr

References

C. Brouard, F. d'Alché Buc, and M. Szafranski.
Semi-supervised penalized output kernel regression for link prediction.
In *International Conference on Machine Learning*, pages 593–600, 2011.

A. Caponnetto, C. A. , M. Pontil, and Y. Ying.
Universal multitask kernels.
J. Mach. Learn. Res., 9, 2008.

F. Dinuzzo and K. Fukumizu.
Learning low-rank output kernels.
In *Proceedings of the 3rd Asian Conference on Machine Learning*, volume 20 of *JMLR: Workshop and Conference Proceedings*, November 2011.

N. Lim, F. d'Alché Buc, C. Auliac, and G. Michailidis.
Operator-valued kernel-based autoregressive model with application to biological network inference, under submission, 2012.

N. Lim, Y. Senbabaoglu, F. d'Alché Buc, C. Auliac, and G. Michailidis.
Okvar-boost: a novel boosting algorithm to infer nonlinear dynamics and interactions in gene regulatory networks, under submission, 2012.

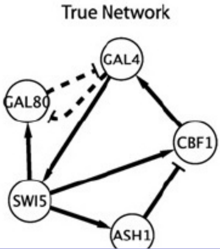
C. A. Micchelli and M. A. Pontil.
On learning vector-valued functions.
Neural Computation, 17:177–204, 2005.

R.J. Prill, D. Marbach, J. Saez-Rodriguez, P.K. Sorger, L.G. Alexopoulos, Xue X., ND Clarke, Altan-Bonnet G, and Stolovitzky G.
Towards a rigorous assessment of systems biology models: The dream3 challenges.
PLoS ONE, 5(2):e9202, 2010.

E. Senkene and A. Tempel'man.
Hilbert spaces of operator-valued functions.
Lithuanian Mathematical Journal, 13(4), 1973.

IRMA

- IRMA stands for In vivo Reverse-engineering And Modeling Assessment
- Cantone et al. built a synthetic network in yeast *S. cerevisiae*
- In this network, each of the 5 genes controls transcription of at least another gene
- Galactose and glucose are respectively used to switch on or off the network.
- Time-series measurements with 16 time-points ("switch off" series) and 21 time-points ("Switch on" series).



IRMA

	Switch-on $\lambda_1 = 10^{-1}$	Switch-on $\lambda_2 = 10^{-1}$	Switch-off $\lambda_1 = 10^{-1}$	Switch-off $\lambda_2 = 10^{-1}$
W	AUROC	AUPR	AUROC	AUPR
Truth	0.932	0.712	0.814	0.754
PC	0.758	0.271	0.482	0.384
1_H	0.531	0.384	0.719	0.667